REMARKS

This amendment is responsive to the Office Action[1] mailed on December 15,

2005. Claims 1-89 were presented for examination and were rejected. No claims are

amended, added or canceled. Thus, claims 1-89 are pending. Claims 1, 12, 24, 37, 49,

52, 59, 63, 68, 70, 75 and 81 are independent claims.

Claims 1-89 are rejected under 35 U.S.C. § 102(b) as being anticipated by Chen et

al., "A C++Data Model Supporting Reachability Analysis and Dead Code Detection"

(hereinafter "Chen"). Applicants respectfully traverse this rejection for the following

reasons.

MPEP § 2131 states that to anticipate a claim, the reference must teach every

element of the claim. "A claim is anticipated only if each and every element as set forth

in the claim is found, either expressly or inherently described, in a single prior art

reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2

USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as

complete detail as is contained in the ...claim." *See Richardson v. Suzuki Motor Co.*, 868

F. 2d 1226, 1236, 9USPQ2d 1913, 1920 (Fed. Cir. 1989). It is respectfully submitted that

Chen does not satisfy this requirement because each and every element of Applicants'

---

[1] The Office Action may contain a number of statements characterizing the cited references and/or the claims which Applicants may not expressly identify herein. Regardless of whether or not any such statement is identified herein, Applicants do not automatically subscribe to, or acquiesce in, any such statement. Further, silence with regard to rejection of a dependent claim, when such claim depends, directly or indirectly, from an independent claim which Applicants deem allowable for reasons provided herein, is not acquiescence to such rejection of that dependent claim, but is recognition by Applicants that such previously lodged rejection is moot based on remarks and/or amendments presented herein relative to that independent claim.

claims (1) cannot be found in Chen, and (2) cannot be found in Chen in any detail, much

less in as complete detail as is contained in the claim, for the following reasons.

Chen is a scholarly IEEE publication which is directed primarily to the computer

science subjects of "Reachability Analysis" and "Dead Code Detection." Chen is not

directed to Applicants' claimed subject matter of, for example, management software

written in a first computer language compatible with first software architecture and not

compatible with second software architecture, as recited in many of Applicants' claims.

Reachability analysis can process each reachable state of a data model, thereby

determining which actions can be performed, and can extend the set of reachable states

with the states generated by performing the actions. Dead code detection relates to the

discovery of unnecessary, inoperative code, such as, for example, dead procedure, dead

variable, dead parameter, dead return value, dead event declaration, dead enumeration or

constant, dead user-defined type, dead class, dead interface, dead module, dead control

and dead external declaration. Neither one of these two subjects is germane to the subject

matter of Applicants' claims, wherefore the general thrust of Chen is in a direction other

than in the direction of Applicants' claimed subject matter.

More specifically, <u>Chen has nothing to do with the subject of incompatible</u>

<u>architectures. Chen teaches conversion of software for the purpose of analysis, not for</u>

<u>the purpose of integration between new and legacy architectures as reflected in</u>

<u>Applicants' specification, drawings and claims</u>.

Moreover, there may be a misunderstanding reflected in the Office Action. The

Office Action, page 2, bottom, says:

> "Given the broadest reasonable interpretation of followed claims in light
> of the specification. As per claim 1: Note: The specification refers CIM or XML

3

as a different computer language from the first computer language such as C++.
However the definition of CIM is 'Common Information Model', and XML
document file that is viewable by a browser."

Although this statement is vague and indefinite, and although Applicants cannot fully

comprehend what is intended by this statement, Applicants believe that the Examiner

may be attempting to allege a commonality between C++ and XML. If that is what the

above-quoted Office Action language is intended to mean, then Applicants respectfully

disagree. In fact, C++ and XML are not inherently compatible as clearly explained in

Applicants' specification, for example with respect to Figs. 2A and 2B, and to that extent

there is no commonality there-between. Indeed, C++ falls within the category of legacy

architecture while CIM/XML architecture is new, non-legacy architecture. These two

architectures are, by definition, mutually incompatible. Simply because CIM uses the

word "Common" in its name, that does not mean that CIM/XML is, necessarily, common

to all architectures including legacy architecture such as C++ or RAID++. However,

since Applicants are not fully certain of what the point of the above-quoted Office Action

language is, Applicants may not have addressed that point herein. If the following

remarks are not deemed sufficiently persuasive for the Examiner to withdraw the Chen

reference, then Applicants request clarification of the above-quoted language in the next

office action.

Consider claim 1, for example:

A computer system employing management software written in a
first computer language compatible with first software architecture and not
compatible with second software architecture, said system comprising: a *schema*
formed within said first software architecture; header files contained within said
schema, said header files being represented in said first language and capable of
being utilized by said management software; means for manipulating said header
files to locate public functions and/or data attributes of said header files; means,
responsive to operation of said manipulating means, for emitting code that calls

said public functions and/or data attributes in said first language to obtain called public functions and/or data attributes; and, means for converting said called public functions and/or data attributes to representations of said called public functions and/or data attributes formed in a different computer language compatible with said second software architecture. (Emphasis added.)

It is clear that Applicants' header files are contained within the schema which is formed within the first software architecture. But, Applicants' first software architecture is legacy architecture with which the recited "management software written in a first computer language" is compatible. This is clear from the dependent claims. For example, claim 3, dependent from claim 1, defines the first computer language as RAID++ and the different computer language as XML/CIM. From Applicants' Fig. 2A, for example, it is clear that RAID++ is legacy architecture and XML is the new, standard architecture.

Accordingly, Applicants submit that the Office Action erroneously applies Chen against Applicants' recited "schema." On page 3 of the Office Action, it states: "*a schema formed within said first software architecture* (See p. 690, left col., first bullet);" (emphasis in original). The first bullet highlights "schema." However, this first bullet is one of five bullets (the other four highlighting "database view", "source view", "graph view", and "GUI front end") which are characterized as five sections that are included in a "specification file."

In the immediately-previous paragraph in Chen, it states:

The query and visualization subsystem of Acacia is built by constructing a C++ instance of the CIAO system [6] using an *instance compiler* that takes a **specification file for a new language** or document type and generates the complete query and visualization environment automatically.

(Italics in original, bold/underline emphasis added.) Therefore, the specification file

which includes software functionality characterized by these five bullets including the

schema is a <u>new language</u> specification file in Chen. A new language specification file is

not, and cannot be, a legacy language file, by definition.

But, as Applicants have shown above, the recited first software architecture

clearly means legacy architecture. Since Applicants' claims clearly recite "a schema

formed within said first software architecture" Applicants' recited schema is formed

within legacy architecture, not in a new language, or in a specification file for a new

language, as is the schema described in Chen. Therefore, this section of Chen to which

the Examiner refers, or anyplace else in Chen does not disclose or suggest "a schema

formed within said first software architecture" as recited in Applicants' claim 1. Since

MPEP § 2131 states that to anticipate a claim, the reference must teach every element of

the claim, the 35 U.S.C § 102(b) rejection of claim 1 should be withdrawn and the claim

allowed <u>for this reason alone</u>.

Furthermore, Applicants submit that the Office Action's association of page 683,

left. col., "A C++ DATA MODEL" in Chen with the various remaining claim elements

of claim 1 (see Office Action, page 3) is an insufficient basis upon which to reject claim

1. In the first place, none of the subject matter recited in any of those claim elements

even appears to be disclosed or suggested in that section of Chen, or elsewhere in Chen.

However, *arguendo*, assuming that such an association was not completely irrelevant on

its face (as it appears to be), the rejection of those claim elements would still be without

merit because the schema in Chen is not equivalent to Applicants' recited schema for

reasons given above. The fact that Chen's schema is different from Applicants' schema

creates a difference that *ripples* through the remaining claim elements of claim 1.

For example, Applicants recite "header files contained <u>within said schema</u>..."

(emphasis added) wherefore a header file disclosed in Chen, that may be within the

disclosed schema of Chen, if any, cannot be viewed as being equivalent to Applicants'

recited header files contained within "said schema", because Chen's disclosed schema is

different from Applicants' schema. That difference flows from, at least, its position

within non-legacy architecture in Chen but within legacy architecture (first software

architecture) in Applicants' claim 1. Therefore, Chen's header file cannot be equivalent

to Applicants' recited header file wherefore Chen does not disclose or suggest "header

files contained within <u>said</u> schema, said header files being represented in said first

language and capable of being utilized by said management software" as recited in claim

1 (emphasis added). Accordingly, the 35 U.S.C § 102(b) rejection of claim 1 should be

withdrawn and the claim allowed <u>for this additional reason alone</u>.

For another example, Chen cannot disclose Applicants' "header files

manipulating means" because, as just noted, Chen's header file is not equivalent to

Applicants' header file. Thus, any manipulating of a header file in Chen would be a

manipulation of a non-equivalent header file. Therefore, Chen cannot disclose or

suggest: "means for manipulating <u>said</u> header files to locate public functions and/or data

attributes of said header files" as recited in claim 1 (emphasis added). Accordingly, the

35 U.S.C § 102(b) rejection of claim 1 should be withdrawn and the claim allowed <u>for</u>

<u>this additional reason alone</u>.

Similarly, Chen cannot disclose or suggest Applicants' recited code emitting means because, as just noted, Chen's "header files manipulating means", if any, would not be equivalent to Applicants' header files manipulating means. Any code emitted in Chen that calls public functions, etc., if any, would be responsive to operation of a non-equivalent manipulating means. Therefore, Chen cannot disclose or suggest: "means, responsive to operation of <u>said manipulating means</u>, for emitting code that calls said public functions and/or data attributes in said first language to obtain called public functions and/or data attributes" as recited in claim 1 (emphasis added). Accordingly, the 35 U.S.C § 102(b) rejection of claim 1 should be withdrawn and the claim allowed <u>for this additional reason alone</u>.

Finally, the schema difference ripples through the last element of claim 1. Applicants' converting means, likewise cannot be disclosed or suggested by Chen because, as just noted, Chen's code emitting means, if any, is not equivalent to Applicants' recited code emitting means. Thus, the called public functions and/or data attributes, if any, called by Chen's code emitting means, if any, are not equivalent to Applicants' called public functions and/or data attributes. Therefore, Chen cannot disclose or suggest: "means for converting <u>said</u> called public functions and/or data attributes to representations of said called public functions and/or data attributes formed in a different computer language compatible with said second software architecture" as recited in claim 1. Accordingly, the 35 U.S.C § 102(b) rejection of claim 1 should be withdrawn and the claim allowed <u>for this additional reason alone</u>.

Independent claims 12, 24, 37, 49 and 52 each recite schema or header subject matter similar to that recited in claim 1, as follows, (with emphasis added):

8

Claim 12 recites, *interalia*: "A computer network employing a computer system utilizing *management software written in a first computer language compatible with first software architecture* and not compatible with second software architecture, said network comprising: a *schema formed within said first software architecture; header files contained within said schema*, said header files being represented in said first language and capable of being utilized by said management software."

Claim 24 recites, *interalia*: "A method for utilizing standardized software architecture to be practiced in a computer system employing *management software written in a first computer language compatible with first software architecture and not compatible with said standardized software architecture*, said method comprising: *said management software utilizing a schema having header files in said first language*."

Claim 37 recites, *interalia*: "A computer program product including *management software written in a first language* and embodied in a computer-readable medium *for operation on a computer system designed in accordance with first software architecture and not compatible with other than said first software architecture*, said computer program product comprising: programmable code for utilizing *a schema having header files in said first language*."

Claim 49 recites, *interalia*: "A computer program product compatible with preferred non-legacy software architectures and operating in a computer system employing management software written in *a first computer language compatible with legacy software architecture* and not compatible with said preferred non-legacy software architectures, said computer program product embodied in a computer readable medium

and comprising: programmable code for utilizing *a schema having header files in said first language.*"

Claim 52 recites, *interalia*: "In a computer network including a computer system having a functional system therein with management software including *a schema for managing said functional system under control of said computer system in accordance with first software architecture*, a translator-compiler for permitting communication about said managing said functional system to be transmitted between said computer system and computer devices operating under second software architecture, said translator-compiler comprising: program code for accessing *header files within said schema* to obtain a header file containing particular information."

Claim 59 recites, interalia: "In a computer network including a computer system and a functional system controlled by said computer system, *management software compatible with legacy software architecture having header files*, said management software being deployed on both said computer system and said functional system, said *management software comprising: translator software means for receiving and manipulating said header files.*"

Claim 75 recites interalia: "In a computer network including a computer system and a storage system controlled by said computer system, a method for managing storage compatible with *software architecture having header files*, said method being deployed on both said computer system and said storage system, said method comprising: translating and manipulating said header files; receiving first requests from outside of said network in first language *incompatible with said software architecture.*"

As can be seen, in each of these independent claims, the header files are included in a schema in a first language and/or in a first or legacy architecture. As such, they are not equivalent to the header files in Chen which are characterized as being included in a new language specification file. It is respectfully submitted that the 35 U.S.C.§ 102(b) rejection of claims 12, 24, 37, 49, 52, 59 and 75 should be withdrawn and the claims allowed, at least for the reasons given above with respect to claim 1.

Claims 2-11, dependent from claim 1, are allowable, at least for reasons based on their dependency, directly or indirectly, from allowable base claim 1.

Claims 13-23, dependent from claim 12, are allowable, at least for reasons based on their dependency, directly or indirectly, from allowable base claim 12.

Claims 25-36, dependent from claim 24, are allowable, at least for reasons based on their dependency, directly or indirectly, from allowable base claim 24.

Claims 38-48, dependent from claim 37, are allowable, at least for reasons based on their dependency, directly or indirectly, from allowable base claim 37.

Claims 50-51, dependent from claim 49, are allowable, at least for reasons based on their dependency, directly or indirectly, directly or indirectly, from allowable base claim 49.

Claims 53-58, dependent from claim 52, are allowable, at least for reasons based on their dependency, directly or indirectly, from allowable base claim 52.

Claims 60-62, dependent from claim 59, are allowable, at least for reasons based on their dependency, directly or indirectly, from allowable base claim 59.

Claims 76-80, dependent from claim 75, are allowable, at least for reasons based on their dependency, directly or indirectly, from allowable base claim 75.

According to the Office Action, independent claims 63, 68, 70 and 81 are each

rejected on the basis of the rationale used to reject claim 1. The rejection of claim 63 also

relies upon previously unapplied Section VI Web-based/Web Service. The rejection of

claims 70 and 81 also relies on Section B, page 690 and previously unapplied Section VI,

page 692.

These independent claims do not recite schema or header subject matter. It is

therefore assumed that page 690, left column, first bullet (Schema) is not being applied to

these claims. This leaves the remaining sections that were applied against claim 1,

namely: page 683, left column, "A C++ DATA MODEL" and page 690, left column,

Section B referring to an instance compiler and discussion of a document file, along with

the new sections mentioned above, as allegedly being applicable, variously, against these

claims. It is respectfully submitted that there is no appropriate rationale used with respect

to the rejection of claim 1, and that there is no description within section VI in Chen, to

read on:

- the receiving, obtaining and converting steps of claim 63;

- the receiving, obtaining and converting programmable code of claim 68;

- the receiving, obtaining and converting steps of claim 70; and

- the interface of claim 81.

To be an effective 35 U.S.C. § 102(b) reference, not only must that reference show each

and every claim element, but it must do so in as complete detail as is contained in the

claim. Applicants submit that, in this instance, the relied-upon sections of Chen do not

show each and every claim element of these four claims. Indeed, they do not show any

of the claim elements of these four claims. And, even if they did (which they do not), there would be no detail given with respect to that showing of those claim elements, much less giving complete detail as is contained in the claim. This is not surprising because, as previously mentioned, Chen has nothing to do with the subject of incompatible architectures. Chen teaches conversion of software for the purpose of analysis, not for the purpose of integration between new and legacy architectures as reflected in Applicants' specification, drawings and claims. There simply is no discussion of incompatible architectures in these sections of Chen, or anywhere else in Chen.

It is therefore submitted that the rejection of claims 63, 68, 70 and 81 under 35 U.S.C § 102(b) be withdrawn and the claims allowed.

Claims 64-67, dependent from claim 63, are allowable, at least for reasons based on their dependency, directly or indirectly, from allowable base claim 63.

Claim 69, dependent from claim 68, is allowable, at least for reasons based on its dependency from allowable base claim 68.

Claims 71-74, dependent from claim 70, are allowable, at least for reasons based on their dependency, directly or indirectly, from allowable base claim 70.

Claims 82-89, dependent from claim 81, are allowable, at least for reasons based on their dependency, directly or indirectly, from allowable base claim 81.

If the Examiner does not find this segment of this Reply to be sufficiently persuasive to withdraw Chen as an applied reference against claims 63, 68, 70 and 81, then Applicants respectfully insist that the next office action point to SPECIFIC

LANGUAGE in Chen, including within the currently relied-upon sections, to read on

EACH and EVERY one of the various bulleted claim elements enumerated above.
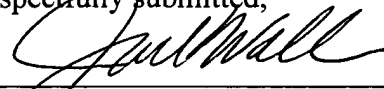
CONCLUSION

In view of the above arguments, the rejection of claims 1-89 should be withdrawn. Reconsideration and allowance are respectfully requested.

To the extent that the above-discussed, or any other, Office Action citations of Chen were applied against particular dependent claim elements but not expressly rebutted herein, it is to be understood that Applicants' silence does not mean or imply acquiescence. Rather, Applicants believe that any response to application of such citations would be moot in view of the foregoing arguments and provisions of MPEP §§ 2131.

To the extent that an extension of time may be needed in order to enter this Reply in this case, please consider this response as including a petition under 37 C.F.R. § 1.136 for such extension of time. Please charge any fee for such petition or any other fee or cost that may be incurred by way of this amendment to Patent Office deposit account number **05-0889**.

If the Examiner feels that a telephone conversation may serve to advance the prosecution of this application, he is invited to telephone Applicants' undersigned representative at the telephone number provided below.

Respectfully submitted,

JOEL WALL
Registration No. 25,648

TELEPHONE:
JOEL WALL ESQ.
508-625-1323
March 10, 2006

15